NJ Department of Education
Office of Educational Technology
Digital Learning NJ (DLNJ)

Grade Band Resources to Support Standard 8.2, Strand E: Computational Thinking

Strand	**E. Computational Thinking: Programming:** *Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.*		
Grade Level bands	**Content Statement Students will be able to understand:**	**Indicator**	**Indicator**
K-2	**Computational thinking and computer programming as tools used in design and engineering.**	**8.2.2.E.1**	List and demonstrate the steps to an everyday task.
		8.2.2.E.2	Demonstrate an understanding of how a computer takes input through a series of written commands and then interprets and displays information as output.
		8.2.2.E.3	Create algorithms (a sets of instructions) using a pre-defined set of commands (e.g., to move a student or a character through a maze).
		8.2.2.E.4	Debug an algorithm (i.e., correct an error).
		8.2.2.E.5	Use appropriate terms in conversation (e.g., basic vocabulary words: input, output, the operating system, debug, and algorithm).
	Resources to support students' progress on the K-2 grade span indicators		**INTRODUCTORY CODING (See more coding resources at the end of this document)** Code.org: • https://code.org/educate/k5 • https://code.org/educate/k5-district-partnership • https://code.org/professional-development-workshops

			Scratchjr: http://www.scratchjr.org/ CS Unplugged: **http://csunplugged.org/** App by codeSpark: http://thefoos.com/ Google's Blockly Games: https://blockly-games.appspot.com/?lang=en **ARTICLES** Article discusses resources for home and school: https://www.yahoo.com/tech/how-to-turn-your-kid-into-a-coder-214022173.html Article discusses teaching strategies: http://www.edutopia.org/blog/15-ways-teaching-students-coding-vicki-davis **BASIC COMPUTER-RELATED VOCABULARY WORDS (source: Google unless otherwise specified)** **Input**-verb: put (data) into a computer. synonyms: feed in, put in, load, insert; **Output**-verb: produce, deliver, or supply (data) using a computer or other device. Example: "you can output the image directly to a video recording system" **Operating system**-noun; the software that supports a computer's basic functions, such as scheduling tasks, executing applications, and controlling peripherals. **Debug**- verb: identify and remove errors from (computer hardware or software). Example: "games are the worst to debug"; noun: the process of identifying and removing errors from computer hardware or software. **Algorithm** - noun: algorithm; plural noun: algorithms-a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. Example: "a basic algorithm for division"
3-5	**Computational**	**8.2.5.E.1**	Identify how computer programming impacts our everyday lives.

	thinking and computer programming as tools used in design and engineering.	**8.2.5.E.2**	Demonstrate an understanding of how a computer takes input of data, processes and stores the data through a series of commands, and outputs information.
		8.2.5.E.3	Using a simple, visual programming language, create a program using loops, events and procedures to generate specific output.
		8.2.5.E.4	Use appropriate terms in conversation (e.g., algorithm, program, debug, loop, events, procedures, memory, storage, processing, software, coding, information, and data).
	Resources to support students' progress on the 3-5 grade span indicators		**CODING (See more coding resources at the end of this document)** Code.org: • https://code.org/educate/k5 • https://code.org/educate/k5-district-partnership • https://code.org/professional-development-workshops Scratchjr: http://www.scratchjr.org/ CS Unplugged: **http://csunplugged.org/** **ARTICLE** Discusses resources for home and school: https://www.yahoo.com/tech/how-to-turn-your-kid-into-a-coder-214022173.html **FREE PROGRAM** CS First increases student access and exposure to computer science (CS) education through after-school, in-school, and summer programs: http://www.cs-first.com/ **WORKING DEFINITIONS (SOURCE: GOOGLE UNLESS OTHERWISE SPECIFIED)** **Algorithm** - noun: algorithm; plural noun: algorithms-a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. Example: "a basic algorithm for division" **Program-** verb: provide (a computer or other machine) with coded instructions for the automatic performance of a particular task. Example: "it is a simple matter to program the

		computer to recognize such symbols" **Debug**- verb: identify and remove errors from (computer hardware or software). Example: "games are the worst to debug"; noun: the process of identifying and removing errors from computer hardware or software. **Loop** – noun: a structure, series, or process the end of which is connected to the beginning. **Events**-An event, in a computing context, is any identifiable occurrence that has significance for system hardware or software. User-generated events include keystrokes and mouse clicks, among a wide variety of other possibilities. System-generated events include program loading and errors, also among a wide variety of other possibilities. **Procedures-**a set of instructions that performs a specific task; a subroutine or function. (source: http://www.yourdictionary.com/procedure#YfFzgp8fEa0T3lWm.99) **Memory**-computer hardware devices used to store information for immediate use in a computer; it is synonymous with the term "primary storage". (source: www.wikipedia.com) **Storage**-computer data storage, often called storage or memory, is a technology consisting of computer components and recording media used to retain digital data. **Processing**-verb: operate on (computer data) by means of a program. **Software**–noun: the programs and other operating information used by a computer. **Coding**-The symbolic arrangement of statements or instructions in a computer program in which letters, digits, etc. are represented as binary numbers; the set of instructions in such a program. Example: That program took 3000 lines of code. (source: dictionary.com) **Information**-when information is entered into and stored in a computer, it is generally referred to as data. After processing (such as formatting and printing), output data can again be perceived as information. **Data**- the quantities, characters, or symbols on which operations are performed by a

			computer, being stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media.
6-8	**Computational thinking and computer programming as tools used in design and engineering.**	**8.2.8.E.1**	Identify ways computers are used that have had an impact across the range of human activity and within different careers where they are used.
		8.2.8.E.2	Demonstrate an understanding of the relationship between hardware and software.
		8.2.8.E.3	Develop an algorithm to solve an assigned problem using a specified set of commands and use peer review to critique the solution.
		8.2.8.E.4	Use appropriate terms in conversation (e.g., programming, language, data, RAM, ROM, Boolean logic terms).
	Resources to support students' progress on the 6-8 grade span indicators		**CODING (See more coding resources at the end of this document)** Code.org has partnered with Bootstrap to develop curricula which teaches concepts in the following content area through computer programming: https://code.org/curriculum/algebra https://code.org/curriculum/science https://code.org/educate/professional-development • MIT App Inventor: http://appinventor.mit.edu/explore/ • Scratchjr: https://scratch.mit.edu/ • CS Unplugged: http://csunplugged.org/ • Birdbrain Technologies: http://www.finchrobot.com/ **FREE PROGRAM** CS First increases student access and exposure to computer science (CS) education through after-school, in-school, and summer programs: http://www.cs-first.com/ **ROBOTICS** • Educational robots for the absolute beginner: https://cs4hsrobots.appspot.com/course

			• Carnegie Mellon Robotics academy: http://education.rec.ri.cmu.edu/ • Bebras CT challenge: http://www.bebraschallenge.org/ **BASIC PROGRAMMING-RELATED TERMS** **Programming**-verb: the action or process of writing computer programs. **Programming language**-noun: A programming language is a formal constructed language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs to control the behavior of a machine or to express algorithms. **RAM**-is an acronym for random access memory, a type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes. RAM is the most common type of memory found in computers and other devices, such as printers. **ROM**-an acronym for read only memory, once data has been written onto a ROM chip, it cannot be removed and can only be read. Unlike main memory (RAM), ROM retains its contents even when the computer is turned off. ROM is referred to as being nonvolatile, whereas RAM is volatile. **Boolean logic terms**- Boolean logic can be used to describe electromagnetically charged memory locations or circuit states that are either charged (1 or true) or not charged (0 or false). The computer can use an AND gate or an OR gate operation to obtain a result that can be used for further processing. (Source: www.techtarget.com)
a9-12	**Computational thinking and computer programming as tools used in design and engineering.**	**8.2.12.E.1**	Demonstrate an understanding of the problem-solving capacity of computers in our world.
		8.2.12.E.2	Analyze the relationships between internal and external computer components.
		8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
		8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI,

			abstraction, variables, data types and conditional statements).
	Resources to support students' progress on the 9-12 grade span indicators		**CODING** • Exploring Computer Science: http://www.exploringcs.org/ • Code.org: https://code.org/educate/csp • Made with code: https://www.madewithcode.com/ • Khan academy: https://www.khanacademy.org/computing/computer-programming **CURRICULUM** • PLTW Computer Science Curriculum: https://www.pltw.org/our-programs/computer-science/computer-science-curriculum **AP COURSES** • AP Course for Computer Science: http://apcsprinciples.org/ • AP Computer Science Principles: https://advancesinap.collegeboard.org/stem/computer-science-principles • AP Computer Science Course: http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html • Free Computer Science Instruction: https://csp-cs4hs.appspot.com/preview • APP Inventor Tutorial: http://appinventor.mit.edu/explore/ • Mobile Computing in APP Inventor: http://mobile-csp.org/ **TEACHER COURSE** CS10K community: https://cs10kcommunity.org/ **FREE PROGRAM**

			CS First increases student access and exposure to computer science (CS) education through after-school, in-school, and summer programs: http://www.cs-first.com/ **ROBOTICS** • Tool that can be programmed: https://www.arduino.cc/ • Educational robots for the absolute beginner: https://cs4hsrobots.appspot.com/course • Bebras CT challenge: http://www.bebraschallenge.org/ • Carnegie Mellon Robotics academy: http://education.rec.ri.cmu.edu/ **GENERAL PROGRAMMING-RELATED TERMS** **Troubleshooting**- troubleshooting is the process or identifying and fixing problems. Computer troubleshooting may involve hardware or software and can sometimes involve both at the same time. The basic process of troubleshooting is to check the most general possible problems first, and then gradually check for more specific problems. (Source: http://pc.net/glossary/definition/troubleshooting) **Peripherals-**a device or unit that operates separately from the CPU but is connected to it. For example a printer. (Source: www.dictionary.com) **Diagnostic software**- Diagnostic software is used to identify problems on a computer or piece of equipment. These programs test the onboard systems for issues and help to alert users of potential problems or breakdowns. (Source: www.wisegeek.com) **Graphic User Interface** (GUI)—software interface designed to standardize and simplify the use of computer programs, as by using a mouse to manipulate text and images on a display screen featuring icons, windows, and menus. (Source: www.dictionary.com) **Abstraction**- abstraction is a technique for managing complexity of computer systems. It works by establishing a level of complexity on which a person interacts with the system, suppressing the more complex details below the current level. (Source: www.en.wikipedia.com) **Variables**- a variable or scalar is a storage location paired with an associated symbolic name (an identifier), which contains some known or unknown quantity of information referred to

			as a value. **Data types**- a particular kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it. **Conditional statements**- A conditional statement, symbolized by p q, is an if-then statement in which p is a hypothesis and q is a conclusion. The logical connector in a conditional statement is denoted by the symbol. The conditional is defined to be true unless a true hypothesis leads to a false conclusion.

GENERAL RESOURCES

Computational Thinking

- Exploring Computational Thinking - Google for Education (130+ resources and videos)
  https://www.google.com/edu/resources/programs/exploring-computational-thinking/

- Computational Thinking in the classroom
  https://computationalthinkingcourse.withgoogle.com/course?use_last_location=true

- CSTA standards
  http://www.csta.acm.org/Curriculum/sub/K12Standards.html

- CSTA Computational Thinking Task Force
  http://www.csta.acm.org/Curriculum/sub/CompThinking.html

- ISTE Computational Thinking for All
  https://www.iste.org/explore/articledetail?articleid=152

- ISTE video on why students need computational thinking
  https://www.youtube.com/watch?v=VFcUgSYyRPg

- Center for Computational Thinking at Carnegie Melon
  http://www.cs.cmu.edu/~CompThink/resources/education.html

- Jeanette Wing's speech about computational thinking
  https://www.youtube.com/watch?v=-CU0hgPmWdY

Coding for Teachers

- Code.org - Online self-paced course for teachers
- https://code.org/educate/professional-development-online

Coding for Students - Lessons on Coding-Computational Thinking

- **Code.org Course 1 for prereaders - grades PreK to 1**
  There are 18 activities, seven of which do not require a computer. There is a video with each unplugged lesson and detailed lesson plans for all the lessons. Every unplugged lesson has handouts for the kids to use.
  https://code.org/curriculum/course1
  - **Grades PK-1 Pre-Reading Examples- Course 1**:
    - Happy Maps - unplugged activity to learn algorithms - Code.org course 1
      https://code.org/curriculum/course1/1/Teacher
    - video overview
      https://www.youtube.com/watch?v=En6Bshuqljg&index=1&list=PL2DhNKNdmOtqBgWyF5kmy2oPh0U-Zfv2G
    - Move-it-Move-it - unplugged activity to give clear instructions - Code.org course 1
      https://code.org/curriculum/course1/2/Teacher#Review
    - video overview
      https://www.youtube.com/watch?v=aV4rqNUG-Jw&feature=youtu.be&list=PL2DhNKNdmOtqBgWyF5kmy2oPh0U-Zfv2G

- **Grades K-5 Code. org- Course 1**:
  - Code.org/K5 - includes a 4 minute video: https://code.org/educate/k5

- **Grades 2-5 Code.org -Course 2**
  There are 19 activities, 8 of which don't require a computer. There is a video with each unplugged lesson and detailed lesson plans for all the lessons. Every unplugged lesson has handouts for students to use. https://code.org/curriculum/course2
  - **Grades K-5 Examples- Course 1**:
    - Binary Bracelets - translating into binary

https://code.org/curriculum/course2/14/Teacher

- video overview - https://www.youtube.com/watch?v=8HANsec7TiQ&index=6&list=PL2DhNKNdmOtobJjiTYvpBDZ0xzhXRj11N

- **K-3 Curriculum & Lessons on Computational Thinking**

  The Foos made by CodeSpark at: http://thefoos.com/comes with a full curriculum which does not require CS knowledge and can be used by teachers or parents. It includes 'unplugged activities' and vocabulary. It can be played on a tablet or browser (not chrome which has disabled some extensions). Students learn Computer Science concepts through a game which does not require reading. The curriculum has 10 lessons that are used by parents or teachers. Each of the lessons correspond to one major concept such as: sequencing, conditional statements, events, loops and debugging. Each lesson activity is a combination of students playing a "game" online with the Foos, class discussion, vocabulary and unplugged activities. The early activities are arranged so that they can be completed by non-readers. Any part of the lesson could be used independently. The software and the introductory curriculum are free.

  Grades K-2 E.1, E.3 and E.4

  Sample lesson: Sequencing a picture story (lesson 1 of Foos curriculum – comes with all handouts, lesson plans and solutions – found at http://thefoos.com/hourofcode/index.html#download_curriculum)

  - Warm up activity – sequence to get ready for school in the morning.
  - Picture story sequence – sequence a picture story and includes debugging – find the picture that does not belong.
  - Includes vocabulary such as programming, algorithm, sequence. It could also be used for E.5.
  - Optional: Allow students access to the Foos software, on any computer that does not run Chrome unless the administrator enables the Unity plug-in.

  Grades 3-5 E.2 and E.3

  Sample lesson using Google CS First. A digital storytelling lesson where students use Scratch to tell a story. Provides videos, lessons, sample solutions materials at http://www.cs-first.com/course/storytelling/video/2692,